Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

System Identification and Control of a Polymer Reactor

Tobias Münker * **Geritt Kampmann** * **Max Schüssler** *
Oliver Nelles *

* *Automatic Control - Mechatronics, University of Siegen, Germany (e-mail: tobias.muenker@uni-siegen.de).*

Abstract: In a polymer production process, a special reactor is used to adjust the viscosity, i.e., chain length of the polymer. This reactor has several control variables mainly in manually control. For future automatic control concepts, such a reactor is modeled from data with a linear (regularized FIR) and a nonlinear state space model (LMSSN). A model predictive control approach is presented in simulation.

Keywords: Reactor control, regularized FIR model, local model state space network (LMSSN), model predictive control (MPC), nonlinear dynamic model, system identification

1. INTRODUCTION

Nowadays, production plants collect vast amounts of process data during production time. Often this data is recorded, and there is an intention to use it for improvements in product quality. This contribution presents a system-based approach for the identification and control of a chemical process. Therefore first, a model is identified and second, a suitable controller for viscosity is developed based on the identified model.

The process data used for this study stems from a continuous *polymer reactor* (PR) equipped with two horizontal agitators. This finishing reactor is used to increase the viscosity of the feed to the desired target. The viscosity corresponds to the mean chain length of the polymer and is one of the key quality figures. Viscosity is mainly controlled by pressure and agitator speed besides other parameters as throughput, catalyst, and temperature. These control parameters are set manually by the plant operators, potentially supported by a viscosity controller acting on the pressure.

The goal of this study is to apply modern techniques from control and identification to improve this situation. In the first step, a novel approach for regularized linear system identification, introduced by Pillonetto et al. (2010), is used to derive a data-based model. In this approach a regularized estimate of the parameters for a finite impulse response model is obtained, see Chen et al. (2012). It has been shown by Pillonetto et al. (2014) in extensive simulation studies that for the identification of dynamic systems, this approach performs favorably. The identification of a linear model allows a thorough system analysis and the use of powerful control techniques. It can be seen that the system has a non-minimum phase behavior. Based on this observation, two control schemes are developed and compared. On the one hand, a classical PI-control is tuned with the identified model. On the other hand, a *model predictive control* (MPC) scheme, see Camacho and Alba (2007), is investigated.

In Sect. 2 both the linear and the nonlinear approach for identification of the plant are described. In Sect. 3 the behavior of the plant is analyzed and two control schemes, one PI control scheme, and a model predictive approach are presented. Section 4 concludes the paper.

2. SYSTEM IDENTIFICATION

The process and the obtained data are described first. Then, the modeling approaches for the linear and nonlinear cases are introduced.

2.1 Process Inputs and Output

The process output and controlled variable is the viscosity denoted by η or y in the following text. In Figs. 1, 4 and 5 the scaled inputs and output of the process are shown. The output is usually kept stationary for a certain production period and only changed if a different viscosity/quality is required. Consequently the inputs show not much excitation during these periods too. As usual, there are also some noise and measurement errors to deal with.

From the available data, eight inputs of the reactor have been identified as most important to describe the viscosity. These are reactor pressure $u_1(k) = p(k)$, agitator speeds $u_2(k) = n_1(k)$ and $u_3(k) = n_2(k)$, flow into the reactor $u_4(k) = \dot{V}(k)$, polymer level $u_5(k) = h(k)$ in the reactor, amount of two catalysts $u_6(k) = c_1(k)$ and $u_7(k) = c_2(k)$, and temperature $u_8(k) = T(k)$ of the polymer before the reactor at discrete time steps k. The sampling time is 6 minutes. The control variables are denoted as $\underline{u}_c(k) = [p(k), n_1(k), n_2(k)]^T$, while the other inputs are chosen in accordance with the desired properties of the end product and cannot be changed by the controller. From a control perspective, these quantities act as measured disturbances on the process. In Fig. 1 the data for the eight inputs used for identification and simulation can be seen. The data was collected during normal operation of the plant and is consequently not optimized for system identification.

Table 1. Increase of model error if given input is not used in the model.

Rank	Name	Input	rel. Error
1	Volume Flow	$\dot{V}$	175%
2	Catalyst 1	c_1	174%
3	Catalyst 2	c_2	142%
4	Temperature	T	114%
5	Speed 2	n_2	113%
6	Pressure	p	108%
7	Speed 1	n_1	104%
8	Polymer Level	h	99%

The importance of the individual inputs for the model was assessed, using the (linear) *finite impulse response* (FIR) model discussed in the next section. The model error with all inputs is compared to models with one of the inputs missing. The results are shown in Tab. 1. The largest influence has the polymer flow $\dot{V}$, increasing the error by 75% when left out, followed by the catalysts. Interestingly, the use of the polymer level h decreases the model quality slightly ($\approx$ 1%), showing that it should be omitted. In

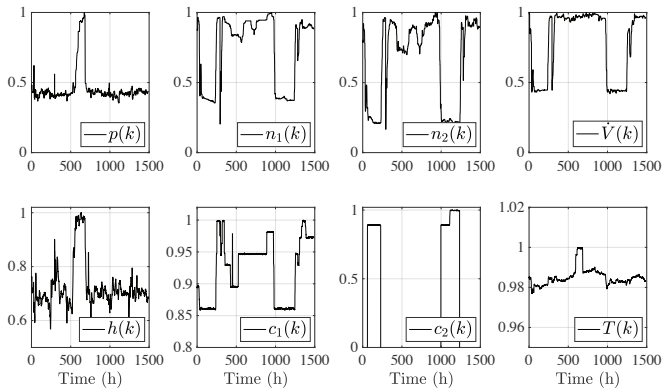

Fig. 1. Process inputs for the viscosity model.

some ranges of the data, the input values are atypical, e.g. inconsistent, out of range or missing. These regions are excluded from the identification by weighting these and the 80 samples (corresponding to the dominating time constant) before by zero.

2.2 Linear Identification

As a first modeling attempt, the impulse responses from the eight inputs to the viscosity η were estimated using a least squares approach. An FIR model represents the convolution by the impulse response explicitly as

$$\hat{y}(k) = \sum_{j=1}^{m} \sum_{i=0}^{n} u_j(k-i) g_j(i)\,, \tag{1}$$

where m is the number of inputs (in our case eight) and n is the length of the impulse response. Its length is chosen to capture relevant dynamic effects and is in our case chosen as 80 with a sampling time of 6 minutes. For the l-th input, the regressor sub-matrix is defined as

$$X_l = \begin{bmatrix} u_l(n+1) & u_l(n) & \dots & u_l(1) \\ u_l(n+2) & u_l(n+1) & \dots & u_l(2) \\ \vdots & \vdots & & \vdots \\ u_l(N) & u_l(N-1) & \dots & u_l(N-n) \end{bmatrix}. \tag{2}$$

These regressor sub-matrices are used to form the complete regressor matrix for the *multiple input - single output* (MISO) case

$$\underline{X} = [\underline{X}_1 \ \underline{X}_2 \ \dots \underline{X}_m]\,. \tag{3}$$

The well-known solution to the least squares estimation of the FIR coefficients can be found by

$$\hat{\underline{\theta}} = \left(\underline{X}^T \underline{X}\right)^{-1} \underline{X}^T \underline{y} \tag{4}$$

with the vector of the measured output values $\underline{y}$. The vector $\underline{\theta}$ contains the estimated impulse response coefficients as

$$\underline{\theta} = \left[g_1(0), \dots, g_1(n), g_2(0), \dots, g_m(n)\right]^T. \tag{5}$$

The step responses identified by this method are shown in Fig. 2. It can clearly be seen that the results are very wiggly and do not look like typical step responses. This problem for FIR models is well known and is caused by the high variance error for the estimation of the coefficients. To overcome this limitation, in Pillonetto et al. (2011) a regularized identification scheme has been proposed which uses the so-called stable spline kernel to regularize the obtained impulse response. Therefore, it is assumed a priori that the impulse response coefficients $\underline{\theta}$ are generated by a random Gaussian distribution with mean zero and covariance $\frac{1}{\lambda}\underline{P}$. This covariance matrix is chosen using the stable spline kernel which is an exponentially warped version of the spline kernel of first order and results in

$$P_{ij} = \alpha^{\max(i,j)} \tag{6}$$

with λ and α being hyperparameters for control of regularization strength and impulse response decay, respectively. For multiple inputs, the regularization matrices are stacked similarly to the complete regression matrix. This results in the formulation of the following optimization problem for the parameters

$$\underset{\underline{\theta}}{\text{minimize}} (\underline{y} - \underline{X}\,\underline{\theta})^T (\underline{y} - \underline{X}\,\underline{\theta}) + \lambda \underline{\theta}^T \underline{P}^{-1} \underline{\theta}. \tag{7}$$

This problem has the solution

$$\hat{\underline{\theta}} = \left(\underline{X}^T \underline{X} + \lambda \underline{P}^{-1}\right)^{-1} \underline{X}^T \underline{y}. \tag{8}$$

Efficient algorithms for the computation of this solution can be found in Chen and Ljung (2013). The values for λ and α are chosen equal for all 8 inputs and are tuned to obtain smooth step responses by hand. The improved step responses for suitable λ are shown in Fig. 3. Important properties of the process can be derived from these responses, e.g., pressure p and flow $\dot{V}$ reduce the viscosity when increased (which was expected from process knowledge), catalyst c_2 has a long delay and level h has non-minimum phase behavior, which decays to zero quickly. Simulation results for training and test data are presented in Fig. 4. The quality of the model is mostly acceptable, but some areas (between 1000 h and 1250 h in Fig. 4) have the potential to significantly improved. Therefore, the nonlinear *local model state space network* (LMSSN) is introduced in the following section.

2.3 Nonlinear Identification

For the approximation of the PR, a nonlinear state space model is used. An advantage of state space approaches (over external dynamics approaches such as the *nonlinear autoregressive model with exogenous input* (NARX)) is the straight forward extension to MISO or *multiple input* -

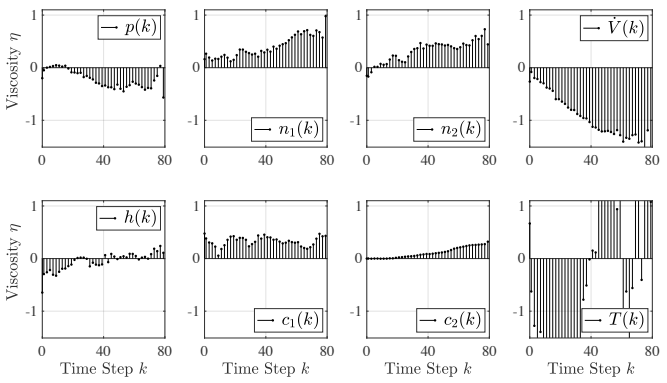

Fig. 2. Estimated step responses of the viscosity η for eight process inputs **without** regularization.

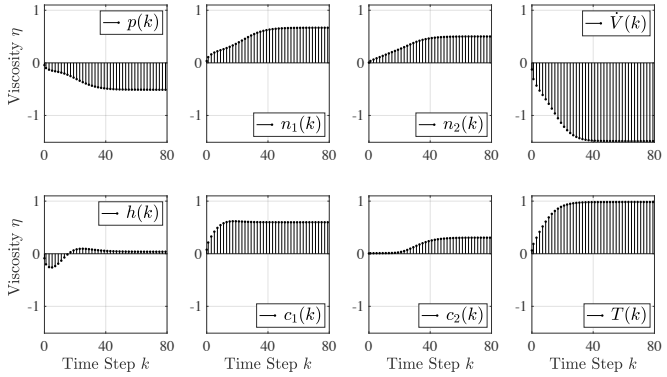

Fig. 3. Estimated step responses of the viscosity η for eight process inputs **with** regularization.

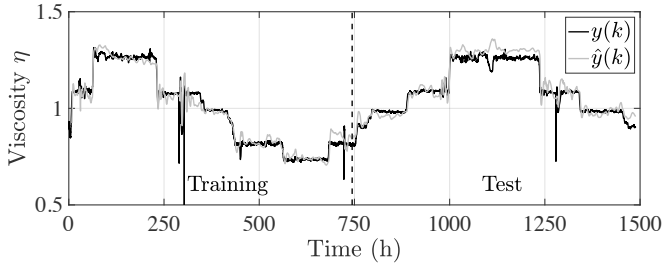

Fig. 4. Training and simulation of a linear model for the viscosity η. NRMSE: 0.3

multiple output (MIMO) models. As multiple inputs are relevant to model the PR's behavior, a MISO nonlinear state space model will be employed. This is described by

$$\underline{\hat{x}}(k+1) = \underline{h}(\underline{u}(k), \underline{\hat{x}}(k)) \tag{9}$$

$$\hat{y}(k) = g(\underline{u}(k), \underline{\hat{x}}(k)), \tag{10}$$

with the state vector $\underline{\hat{x}}(k)$, the input vector $\underline{u}(k) = [u_1(k), u_2(k), \ldots, u_8(k)]^T$, the output $\hat{y}(k)$, the state equations $\underline{h}(\cdot)$, and the output equation $g(\cdot)$ at the discrete time step k. The approximation of the nonlinear functions $\underline{h}(\cdot)$ and $g(\cdot)$ will be done with *local model networks* (LMNs). An LMN, for example for the i-th entry in the state equation, is described by

$$\hat{x}_i(k+1) = \sum_{j=1}^{n_{m_i}} \underbrace{\left(\underline{\tilde{u}}^T(k) \cdot \underline{\theta}_{ij}\right)}_{\text{Local Model}} \overbrace{\Phi_{ij}\left(\underline{\tilde{u}}(k), \underline{c}_{ij}, \underline{\sigma}_{ij}\right)}^{\text{Validity Function}}, \tag{11}$$

where n_{m_i} is the number of local models for LMN output x_i, $\underline{\tilde{u}}^T(k)$ represents the extended input vector [1], $\underline{\theta}_{ij}$ is the parameter vector of an affine local model, and Φ_{ij} is the validity or activation function. The validity function Φ_{ij} is chosen as a normalized *radial basis function* (RBF). Its arguments are also the extended input vector $\underline{\tilde{u}}(k)$, as well as the center coordinates $\underline{c}_{ij}$ of the RBFs, and the standard deviations $\underline{\sigma}_{ij}$ of the RBFs. In this manner, all state equations and the output equation can be modeled by LMNs.

This architecture and the corresponding identification procedure is called *local model state space network* (LMSSN). This method has proven to be very successful on system identification benchmarks, such as shown in Schüssler et al. (2019) and will be, in this contribution, extended to the MISO case for the first time.

The identification algorithm of the LMSSN is based on the *local linear model tree* (LOLIMOT) algorithm developed by Nelles and Isermann (1996) and deeply analyzed in Nelles (2001). LOLIMOT is an incremental tree-construction algorithm that partitions the input space (all dimensions spanned by $\underline{u}(k)$ and $\underline{\hat{x}}(k)$) by axis-orthogonal splits. It constitutes thereby a heuristic way, in which the centers and standard deviations of the RBFs are placed within the input space. This means that only the parameters of the affine local models $\underline{\theta}_{ij}$ need to be optimized. Per iteration, the input space is split once and one further local model is added, increasing the capability of the LMSSN to capture nonlinearities. The initial model is given by the *best linear approximation* (BLA), developed by Pintelon and Schoukens (2012), making sure that all estimated nonlinear models are at least as good as the best linear model. For a detailed account on the LMSSN model and identification procedure refer to Schüssler et al. (2019).

For the identification of an LMSSN model, a data set with $N = 14880$ data points was used with a sampling time of $T_0 = 6\,\text{min}$. The set consists of 8 inputs (see Sec. 2.1) and one output (the viscosity). The dataset is split into three parts: 40 % are used as training data, 20 % as validation data, and the last 40 % of the dataset are used for testing. The algorithm terminates if the error on validation data increases for two consecutive splits.

In total, five LMSSN models with order $n_x = \{1, 3, 5, 7, 9\}$ were estimated. The results on the whole dataset, containing the training, validation, and test sequence are shown for the best LMSSN model with $n_x = 3$ state variables in Fig. 5. The model consists of eight local models (one in the first state equation and output equation each, two in the second state equation, and four in the third state equation). The model has 96 parameters. It can be seen that the model represents the process well.

All estimated models, the number of parameters, and the *normalized root mean square error* (NRMSE) on the test sequence are shown in Tab. 2. The NRMSE is calculated by

[1] $\underline{\tilde{u}}^T(k)$ denotes the input vector for the LMNs, i.e. $\underline{\tilde{u}}^T(k) = [1,\ \underline{u}^T(k),\ \underline{\hat{x}}^T(k)]$ to distinguish it from the dynamic model and process input $\underline{u}(k)$.

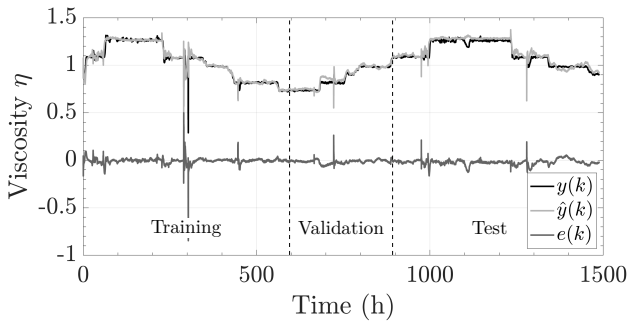

Fig. 5. Output data of process $y(k)$ and model output $\hat{y}(k)$, as well as the model error $e(k)$ for training, validation, and test sequence.

Table 2. Comparison of different LMSSN models, their respective number of parameters and the NRMSE on the test sequence.

States n_x	Number of local models	Number of parameters	NRMSE Test
1	8	80	0.38
3	8	96	0.24
5	13	182	0.42
7	11	176	0.54
9	12	216	0.36

$$e_{\mathrm{NRMS}} = \frac{e_{\mathrm{RMS}}}{\sqrt{\mathrm{Var}(y)}} = \frac{\sqrt{\frac{1}{N}\sum_{k=1}^{N}(y(k)-\hat{y}(k))^2}}{\sqrt{\frac{1}{N}\sum_{k=1}^{N}(y(k)-\bar{y})^2}} \quad (12)$$

where N is the amount of data samples, $y(k)$ is the process output, $\hat{y}(k)$ is the model output and $\bar{y}$ is the mean value of the process output.

With increasing model order, the number of parameters per local model increases. But a higher model order does not necessarily lead to a higher total number of parameters, as the necessary number of local models for a satisfactory model quality is estimated by the LMSSN algorithm. The model with three state variables is chosen, as it has the lowest NRMSE on the test sequence and the number of parameters seems to be reasonable.

3. CONTROL

Based on the identified models from the previous section, a control scheme is derived.

3.1 Reduced Model

First, the behavior of the linear model is analyzed. The behavior of the FIR model is transferred to a state space system with delayed inputs as states. This state space system is then reduced using a balanced truncation, see Skogestad and Postlethwaite (2007). The reactor in the given example has two independently adjustable speeds n_1 and n_2. Since plants are commonly operated this way, these two speeds are set to a fixed (but adjustable) ratio $n_2 = bn_1$, where b is the desired fixed ratio. This results in one single transfer function from speed to viscosity.

The behavior of a sixth-order reduced system for each, reactor pressure p and speed n, is now analyzed. The reduced system is found by a balanced reduction of a state-space model, which represents the SISO impulse response of the respective input. The obtained poles and zeros for the transfer function from p and n to η are shown in Fig. 6.

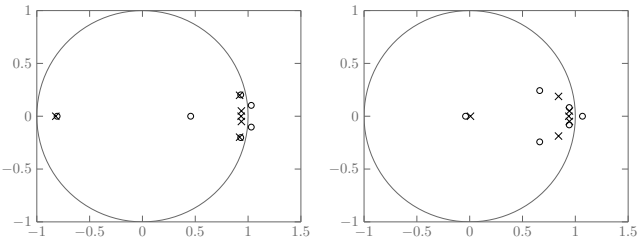

Fig. 6. Pole (x) and zero (o) locations of the reduced order models for transfer functions from p (left) and n (right) to viscosity η.

Here, it can be seen that both transfer functions have zeros lying outside the unit circle. Thus, the system has a non-minimum phase behavior making control more difficult.

3.2 PI-Control

Advanced model-based control laws like *internal model control* (IMC) or MPC are often superior in performance to classical PI-control. However, for practical purposes, a simple design is often advantageous regarding implementation and tuning on-site. A PI-controller is simple, can be implemented in almost all hardware platforms, is well-understood by the operators of a plant and is usually intuitive to tune. The structure of the PI-control scheme

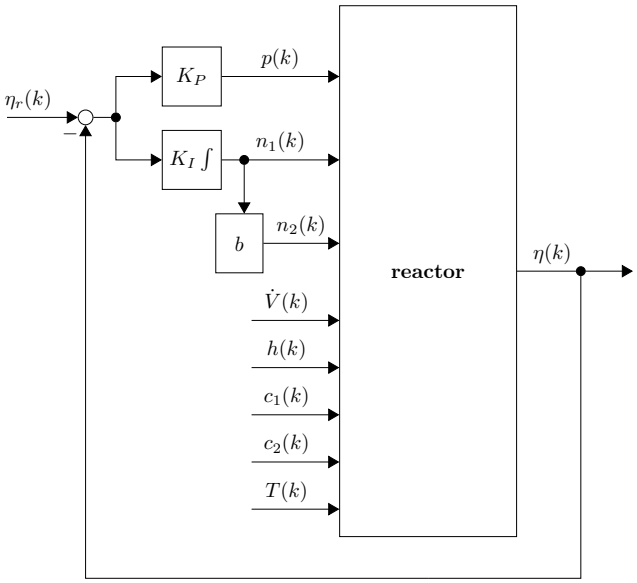

Fig. 7. Divided PI-controller for the polymer reactor.

is shown in Fig. 7.

Here, the P-part of the controller is applied to the pressure and the I-part is applied to the speed. The control structure is referred to as PI-control for simplicity, although it is a P-controller for pressure and an I-controller for speed. This ensures that when the stationary error is zero, the pressure remains at a constant reference value and the

speed is used for adjustment of the set point. First, the behavior of the step response of the closed-loop system is analyzed (see Fig. 8). Here, the actuating variables p, n, the control variable η and the deviation of η from the reference value e_η are shown. The control behavior

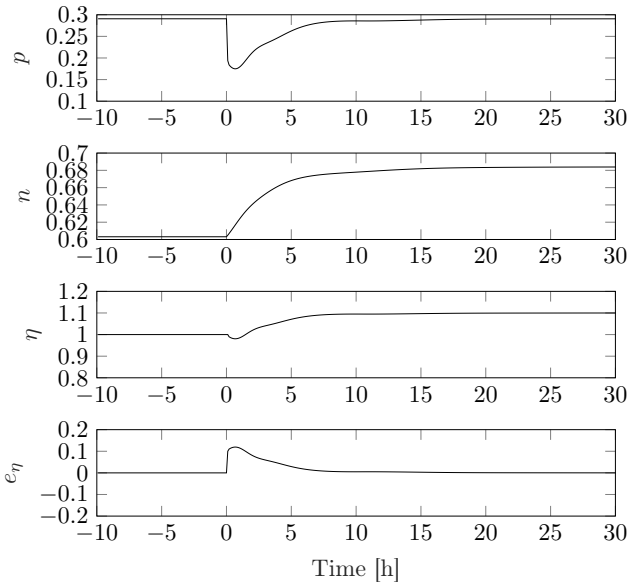

Fig. 8. Reference step response of the proposed PI-control scheme.

is smooth and has a settling time of 10 h. Also, the distur-

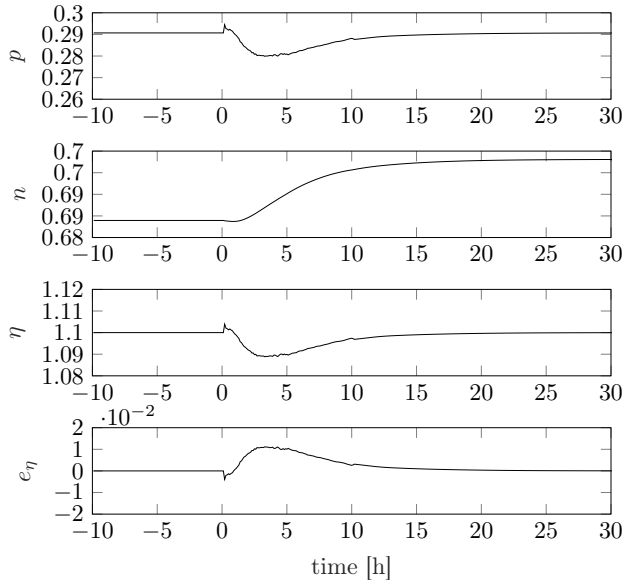

Fig. 9. Disturbance step response from $\dot{V}$ to η, for a step from $\Delta\dot{V}$ of 0.0114 applied at $t = 0$.

bance behavior of the system is analyzed. Therefore, an unmeasured stepwise disturbance at the output is applied. The result is shown in Fig. 9. Here, it can be seen that it takes 12 h to suppress the applied disturbance. The control scheme is able to provide an appropriate stationary behavior, but the dynamic behavior could be improved.

3.3 Model Predictive Control

Model predictive control (see Morari and Lee (1999); Camacho and Alba (2007)) follows a different idea. Therefore in our problem, the following optimization problem is formulated

$$\begin{aligned}
\underset{\underline{\eta},\underline{n},\underline{p}}{\text{minimize}} \quad & \sum_{i=0}^{n_p} \left(\eta_r(k+i) - \eta(k+i)\right)^2 \\
& + \lambda_n \sum_{i=1}^{n_p} \left(n(k+i) - n(k+i-1)\right)^2 \\
& + \lambda_p \sum_{i=1}^{n_p} \left(p(k+i) - p(k+i-1)\right)^2 \\
& + \lambda_{pf} \left(p(k+i) - p_r(k+i)\right)^2 \\
\text{subject to} \quad & \eta(k+i) = \sum_{j=1}^{8} \sum_{l=1}^{n} u_j(k+i-l) g_j(l) + d(k), \\
& i = 1, \ldots, n_p \,.
\end{aligned}$$

The vectors $\underline{\eta}$, $\underline{p}$ and $\underline{n}$ contain n_p future values of η, p and n. The prediction horizon n_p is chosen equal to the order of the identified FIR system. The first term within the optimization problem penalizes the deviation of the viscosity from the set point, while the second and third term penalize the change of the control variables. The higher λ_p and λ_n are chosen the less will the corresponding control variable change. The last term penalizes the deviation of the pressure from a given set point. Since the pressure of the reactor has a relatively limited control range it is used only within the transient phase of a set point change and is penalized to become equal to its reference value by λ_{pf}. The higher λ_{pf} is chosen the faster the pressure is adjusted to its reference value. The disturbance $d(k)$ in the problem formulation is used to mitigate model errors. It is chosen as the model error from the previous time step, but can be extended to more complex noise models. It is well known that this linear MPC problem is a quadratic program and thus convex (Boyd and Vandenberghe (2004)). The problem is solved using the quadprog optimizer in Matlab. First, a reference step is applied to the control system. The response is shown in Fig. 10. A commonly encountered

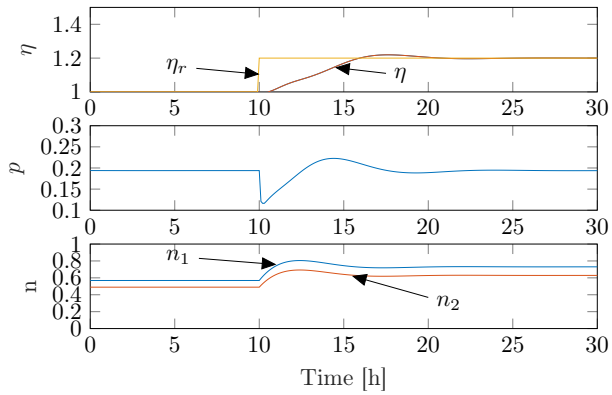

Fig. 10. Reference step response of the MPC. A reference change of η_r from 1.0 to 1.2 is applied at $t = 10$.

mode of operation is the ramp-like change of the volume flow through the reactor. This can be considered as a measured disturbance at the input. In Fig. 11 this behavior

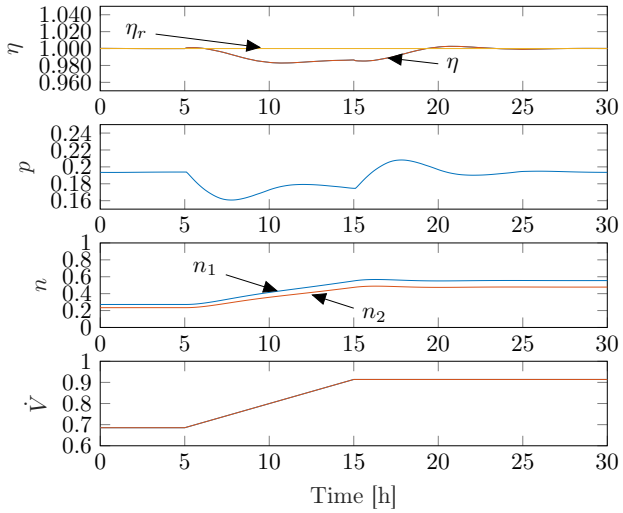

Fig. 11. Ramp-like change of the flow considered as a measured disturbance in the MPC setting.

is shown. It can be seen that although the volume flow is changed, the viscosity is kept tightly at the desired value.

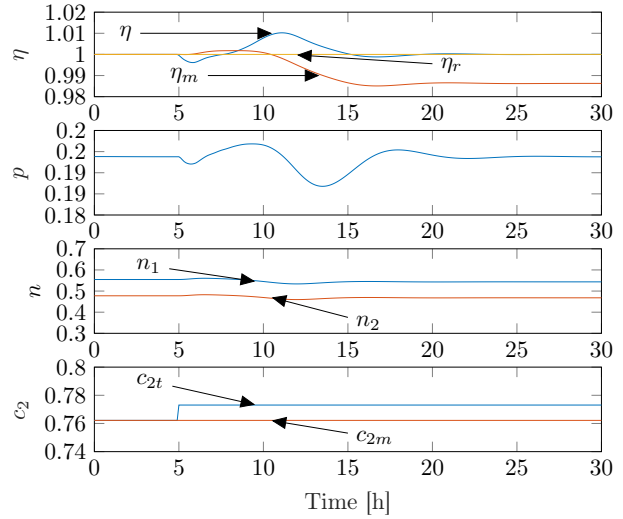

Fig. 12. Stepwise disturbance for the first catalyst. This is an unmeasured disturbance for the MPC problem.

In Fig. 12 another type of disturbance is analyzed, which is usually not measurable. The amount of catalyst applied to the reactor is changed and the reaction of the controller is analyzed. This part of the change has to be derived by the model for $d(k)$. It can be seen that the model value η_m without correction of $d(k)$ is lower than the true value, while the corrected value η coincides with the reference value η_r at the end. The speed is adjusted slightly for the steady-state compensation of the disturbance.

4. CONCLUSION

We demonstrated the effectiveness of statistical data-based techniques for identification and control of a polymer reactor. Therefore, first, a linear system is identified. For its identification, modern kernel based regularization methods are used, which provide significantly better impulse response estimates compared to standard linear models. Furthermore, a nonlinear state space model based on local model networks is identified to capture nonlinear effects within the process. Finally, a model predictive control scheme based on the identified linear model is derived and it is demonstrated that this scheme can suppress both measured and unmeasured disturbances and track the desired reference trajectory. From this, it can be concluded that the combination of advanced identification techniques with model-based control techniques, relying on solutions to optimization problems, will remain an active area of research in the future.

ACKNOWLEDGEMENTS

We thank Technip Zimmer GmbH for the explanations regarding the process and the polymer reactor, and for the provision of the data sets.

REFERENCES

Boyd, S. and Vandenberghe, L. (2004). *Convex optimization*. Cambridge university press.

Camacho, E.F. and Alba, C.B. (2007). *Model predictive control*. Springer Science & Business Media.

Chen, T. and Ljung, L. (2013). Implementation of algorithms for tuning parameters in regularized least squares problems in system identification. *Automatica*, 49(7), 2213–2220.

Chen, T., Ohlsson, H., and Ljung, L. (2012). On the estimation of transfer functions, regularizations and Gaussian processes-revisited. *Automatica*, 48(8), 1525–1535.

Morari, M. and Lee, J.H. (1999). Model predictive control: past, present and future. *Computers & Chemical Engineering*, 23(4), 667–682.

Nelles, O. (2001). *Nonlinear system identification: from classical approaches to neural networks and fuzzy models*. Springer Science & Business Media.

Nelles, O. and Isermann, R. (1996). Basis function networks for interpolation of local linear models. In *Decision and Control, 1996., Proceedings of the 35th IEEE Conference on*, volume 1, 470–475. IEEE.

Pillonetto, G., Chiuso, A., and De Nicolao, G. (2010). Regularized estimation of sums of exponentials in spaces generated by stable spline kernels. In *Proceedings of the 2010 American Control Conference*, 498–503. IEEE.

Pillonetto, G., Chiuso, A., and De Nicolao, G. (2011). Prediction error identification of linear systems: A nonparametric Gaussian regression approach. *Automatica*, 47(2), 291–305.

Pillonetto, G., Dinuzzo, F., Chen, T., De Nicolao, G., and Ljung, L. (2014). Kernel methods in system identification, machine learning and function estimation: A survey. *Automatica*, 50(3), 657–682.

Pintelon, R. and Schoukens, J. (2012). *System identification: a frequency domain approach*. John Wiley & Sons.

Schüssler, M., Münker, T., and Nelles, O. (2019). Local model networks for the identification of nonlinear state space models. In *58th IEEE Conference on Decision and Control (CDC)*. Nice, France.

Skogestad, S. and Postlethwaite, I. (2007). *Multivariable feedback control: Analysis and design*, volume 2. Wiley New York.